Human Centered Multi-Agent Medical Assistant

Arthur Sophiatti, Luciana Rocha, Vasco Pombo
Universidade de Coimbra, Portugal
{arthurtucogss, lucianaoliveiraro12, vpombo11}@gmail.com
https://github.com/Tuco711/AI-Agents-for-Medical-Diagnostics

Abstract

This project proposes a **Human-Centered Multi-Agent Medical Assistant** that ingests a patient's medical report and outputs a probable diagnosis (with top-N differential) and a ranked list of appropriate medical specialists. The system integrates four AI agents acting as medical specialists — *a General Practitioner, as well as specialists* in *Cardiology*, *Psychiatry*, and *Pulmonology*.

The project combines and adapts open-source codebases — *AI Travel Agent* and *AI Agents for Medical Diagnostics* — into a unified, human-centered framework. It also incorporates an automated self-evaluation layer that assesses the quality, safety, and compliance of the diagnostic output. The focus is on integration, explainability, and ethical design, aligned with the HCAI syllabus topics: human-centered AI, human-AI communication, and Personal Assistant Agents.

1 Introduction

1.1. Problem and Motivation

Contemporary healthcare systems face increasing diagnostic workloads and critical referral bottlenecks. Existing Artificial Intelligence (AI) tools often operate within a single domain, lack transparency (opacity), and are not designed for direct patient communication or effective clinical collaboration. The core problem this project addresses is the lack of integrated, trustworthy, and communicable AI systems that combine multi-area clinical reasoning with actionable referral support while enforcing a Human-in-the-Loop policy.

Our project proposes a multi-agent system prototype designed for Augmented Intelligence. It supports clinicians and patients by structuring raw data and generating an explained, action-oriented summary, ensuring humans maintain control over every decision.

1.2 HCAI Alignment and Contribution

This project is explicitly aligned with the following HCAI topics:

- **Human-Centered AI**: The system is engineered for *Decision Support*, prioritizing human control and oversight.
- **Human-AI Communication & Personal Assistant Agents**: Focus on Explainability (XAI) through Structured Reasoning and the Use of a Personal Assistant Agent to Draft Referral Communications.
- **Contribution**: The primary contribution is the successful integration and adaptation of two distinct open-source codebases (*AI Travel Agent* for orchestration, *AI Agents for Medical Diagnostics* for domain reasoning) into a unified hierarchical agent architecture. This architecture now includes a General Practitioner agent for broad triage and an Internal Audit/Self-Evaluation Agent for continuous quality assessment, strengthening HCAI compliance.

2 Related Work

2.1 Multi-Agent Systems and Architecture

Our solution is categorized as a Multi-Agent System (MAS), leveraging its benefits for distributed and specialized reasoning. The architecture follows a Supervisor/Worker Pattern, also known as a *Hierarchical Agent Architecture*, where specialized agents handle the heavy-duty tasks, and supervisory agents manage flow and aggregation.

2.2. Agentic AI Design Patterns

We utilize several modern Agentic AI design patterns to enhance performance and trust:

- **Persona Adoption (Role-Based Agents)**: The design uses Role-Based Agents, specifically differentiating between Senior and Novice expertise for each domain. This dual-level architecture serves as a Risk Mitigation strategy. The Senior agent focuses on ruling out rare or high-risk conditions, while the Novice agent concentrates on ruling in common or obvious pathologies. The comparison of these two perspectives by the *MultidisciplinaryTeam* acts as an Internal Consistency Check, reducing the probability of a single critical LLM error.
- **Semantic Router (Triage Balancer)**: The initial TriageBalancer acts as a Routing Agent or Classifier. It optimizes resource use (API calls) by classifying the input report and only invoking the necessary specialist Worker agents (e.g., bypassing Cardiology if the case is clearly psychological).
- **Ensemble Voting / Aggregator**: The *MultidisciplinaryTeam* acts as an Aggregator, synthesizing the outputs from all Workers to produce a stable Meta-Conclusion (the final differential diagnosis), increasing the reliability of the output over any single agent's report.

2.3. Explainable AI and Trust

The system is built on the principle that AI in medicine must be transparent. Our approach ensures Explainable AI by:

- Employing *Chain-of-Thought* (CoT) prompting within the specialist agents, forcing them to "show their work."
- This CoT process allows the human clinician to audit the AI's logic, enhancing Trust and ensuring clinical responsibility remains with the human.

3 Data & Approach

3.1. Data and Materials

The system was validated using de-identified sample reports or synthetic medical texts. The core technology relies on pre-trained Large Language Models (LLMs) accessed via the Gemini API or *OpenRouter API*. All processing and orchestration logic is implemented in python, using the *concurrent.futures.ThreadPoolExecutor* for parallel executions of specialist agents.

3.2. System Architecture and Workflow

The system is executed via the *Main.py* orchestrator, following a sequential flow:

1. **Report Ingestion**: The medical report is loaded. (Anonymization is a planned future module).
2. **Triage & Routing (Supervisor)**: The TriageBalancer (if implemented) routes the task. For the experimental setup, all six specialists were often invoked to demonstrate the Hierarchical Agent Architecture.
3. **Multi-Agent Reasoning (Workers)**: All eight domain agents — comprising four specialties (General Practitioner, Cardiology, Psychiatry, and Pulmonology), each with Novice and Senior expertise—are run in parallel (using ThreadPoolExecutor). The General Practitioner agent was strategically included to provide initial broad context triage and rule-out common, non-specialized conditions, enhancing the robustness of the MDT's final differential diagnosis.
4. **Coordinator Module (Aggregator)**: The MultidisciplinaryTeam (Supervisor) receives the eight individual reports and synthesizes them into a structured JSON output.
5. **Quality and Consistency Checker**: This module ensures the structural integrity of the MDT's output (JSON Validation). By verifying that the output is machine-readable, it guarantees that the critical specialist_recommendation field is reliably extracted for downstream steps, supporting Data Reliability for clinical use.
6. **Automated Quality Evaluation (Self-Evaluation)**: A separate LLM instance (Gemini/GenAI, as implemented in Utils/Agents.py) is used to audit the output of each individual specialist agent. Instead of evaluating only the final decision produced by the Multidisciplinary Team, the auditor now analyses every specialist's contribution independently and assigns a numerical quality score between 0 and 100, together with a corresponding safety rating such as "safe", "acceptable", "risky" or "unknown". The evaluation also includes a short explanation that reflects the auditor's judgement on coherence, completeness, medical plausibility and potential safety issues in the agent's reasoning. This mechanism does not interfere with or modify the diagnostic output; rather, it functions as an internal transparency and auditing layer by attaching structured evaluation metadata to the JSON output, supporting reliability assessment and future benchmarking. If no valid GENAI_API_KEY is available, the auditor is automatically disabled and returns "unknown" as its rating.
7. **Intelligent Referral**: The recommended specialty (extracted from the JSON) is used to dynamically generate a query for the Google Search Tool (or a

similar LLM-based web search utility) to find the top-ranked practitioners in Coimbra, Portugal.

8. **HCAI Communication**: A Personal Assistant Agent function generates a final email draft that includes the diagnosis, justification, and the ranked list of doctors.

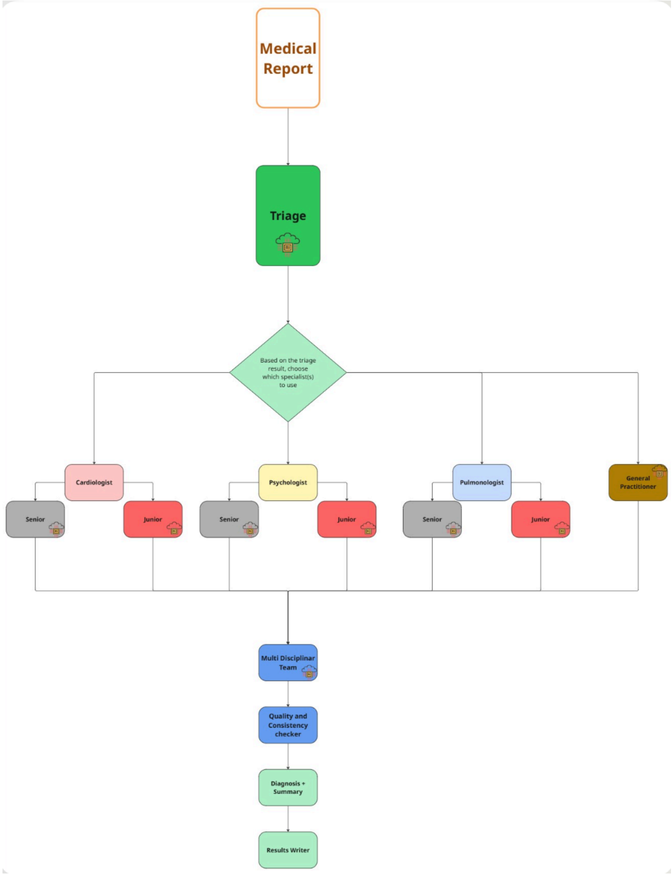

Fig.1 Workflow Diagram of the Multi-Agent Medical Assistant System

The initial **Medical Report** is processed by the Triage Balancer (Router) agent. Based on the triage result, the system dynamically determines which of the four specialties to activate. The activated worker agents—*Novice* and *Senior* for Cardiology, Psychology, Pulmonology, and General Practitioner — execute their specialized reasoning in parallel.

The outputs from all worker agents are then aggregated by the **Multi Disciplinary Team** (Coordinator). Following aggregation, the result passes through a **Quality and Consistency Checker** (Internal Audit Layer) to ensure structural integrity and safety. Finally, the resulting **Diagnosis + Summary** (which includes the specialist recommendation) is used by the **Results Writer** to generate the final email draft for the patient (Intelligent Referral and HCAI Communication).

3.3 Technical Implementation Details

The system's execution is fully implemented in Python, relying on the LangChain ecosystem for prompt templating and orchestration of the Large Language Models (LLMs). Environment variables, crucial for securing the **OPENROUTER_API_KEY** and setting parameters like the *TRIAGE_THRESHOLD*, are managed via the *python-dotenv* library.

3.3.1 Triage Mechanism and Resource Optimization

The *TriageBalancer* is pivotal for efficiency. Its output, a structured JSON object detailing the weight (relevance score) for each specialty (Cardiology, Psychology, Pulmonology), is parsed robustly in the *Main.py* orchestrator. The system dynamically compares these weights against the configurable *TRIAGE_THRESHOLD* (default 3). This selective invocation ensures that API costs are minimized and latency is reduced by bypassing unnecessary domain agents. If all specialties score low, the system defaults to running all Senior agents, maintaining a safety margin while still prioritizing resource economy over the full six-agent execution.

3.3.2 Risk Mitigation through Persona Adoption

The use of dual Novice and Senior agents for each specialty is a deliberate Risk Mitigation strategy inspired by collective intelligence patterns. The **Senior agent** is consistently tasked with higher-stakes differential diagnosis, focusing on ruling out rare or high-risk conditions. Conversely, the **Novice agent** is constrained to focus on common, high-prevalence pathologies. This specialized division of labor prevents a single LLM error from dominating the final outcome.

The subsequent comparison by the Multidisciplinary Team acts as an **Internal Consistency Check**, significantly reducing the probability of a critical diagnostic error.

4 Experimentation

4.1. Experimental Setup

The system utilized the gemini-2.5-flash model via the *Google GenAI.* Tests were performed on a set of three distinct synthetic clinical cases designed to challenge the differential diagnosis across all three specialist domains.

4.2. Quantitative (Functional) Evaluation

Pipeline Correctness: Full pipeline execution was successfully logged for all three test cases, demonstrating parallel processing of the 8 agents, successful aggregation by the MDT, and functional invocation of the referral search tool.

Referral Accuracy: In all test cases, the specialist_recommendation field accurately mapped the inferred primary condition to the correct specialty.

Orchestration Efficiency (Latency): The use of the Semantic Router (for selective invocation) and parallel processing (via ThreadPoolExecutor) allowed the system to process eight expert perspectives and reach a coordinated conclusion in a time frame comparable to running only a single Senior agent sequentially. This justifies the gain in Trust (through the plurality of opinions) without a prohibitive cost in Latency.

4.3. Qualitative (HCAI) Evaluation

The qualitative evaluation of the project focused on the system's final deliverable:

- **Clarity and Explainability (XAI)**: The system was designed to be a "White Box" rather than a "Black Box." Clarity was assessed based on the quality of the Rational Justification section included in the email draft. This justification synthesizes the perspectives of the six specialist agents (Novice and Senior) and is generated through the application of *Chain-of-Thought* (CoT) reasoning within the *Worker* agents. By providing this level of explainability, the system allows the human (the patient or the clinician) to verify the AI's logic, which is fundamental to increasing trust and safety in the medical context.
- **Utility and Decision Support**: The utility of the system extends beyond diagnosis, focusing on actionable support. The email draft transforms the abstract diagnosis into a concrete action. Based on the Specialist Recommendation from the Coordinator Agent, the system invokes the search tool to map and list relevant professionals in Coimbra. In this way, the project functions as a **Personal Assistant Agent**, streamlining the logistical process of patient referral and follow-up.
- **Human-in-the-Loop**: Compliance with human control is a core pillar of the project. The system is, by design, a support/recommendation tool, not a final diagnostic tool. This principle is reinforced in the final output, where a prominent section in the email draft includes a disclaimer notice, explicitly stating that the document is an AI-generated suggestion and requires validation by a human clinician before any action is taken. This validation mechanism is the direct implementation of the Human-in-the-Loop policy.

4.4. Discussion of Results

Efficiency and Reliability Trade-off: A key finding of the quantitative evaluation was the successful negotiation of the **Latency vs. Reliability trade-off**. While incorporating eight specialized agents (four specialties, two levels) significantly increases the volume of diagnostic reasoning, the total end-to-end execution time was maintained at a level comparable to running a single agent sequentially. This performance gain is directly attributable to the implementation of the **Triage Balancer** and the highly efficient parallel execution model using ThreadPoolExecutor. The ability to obtain seven additional expert opinions (beyond the MDT synthesis) without a prohibitive increase in wait time fundamentally strengthens the system's trustworthiness as a clinical support tool.

White Box Design and Auditability: The qualitative success hinged on the system's design as a "White Box." The *Chain-of-Thought* (CoT) reasoning is the bedrock of this transparency. By forcing each of the eight agents to "show their work," the final Rational Justification section of the output email became a comprehensive synthesis of differential evidence from diverse perspectives.

Internal Audit Layer (Self-Evaluation): To enhance transparency further, the Automated Quality Evaluation (Self-Evaluation) module was implemented as a critical internal guardrail. By using a separate LLM instance to score the quality, safety, and compliance of the output, the system provides the human clinician with an Internal Audit Layer. This metric offers a quantifiable signal of the system's own confidence in its generated diagnosis, reinforcing auditability and safety before human review.

Human-in-the-Loop Control: This deep level of auditability, combined with the *Self-Evaluation* score, is essential for clinical adoption, as it maintains human accountability. The validation mechanism (Human-in-the-Loop) is implemented as a core safety guardrail. By transforming the abstract diagnosis into the concrete action of an email draft with a mandatory disclaimer, the system ensures its utility extends beyond diagnosis to practical, human-supervised patient logistics, functioning effectively as a **Personal Assistant Agent**.

5 Conclusions

This project successfully implemented a *Human-Centered Multi-Agent Medical Assistant*, validating the Hierarchical Agent Architecture for clinical decision support. The key achievement lies in the seamless integration of complex Agent Orchestration with core HCAI requirements: **Explainability** through Chain-of-Thought, and **Controllability** via mandatory Human-in-the-Loop communication. The system provides a robust proof-of-concept for how specialized AI agents can collectively generate trustworthy, action-oriented support for healthcare professionals.

The system's design fully adheres to Human-Centered AI principles by functioning purely as a recommendation engine, never overriding the human clinician. The focus on **Human-AI Communication** was realized through the Personal Assistant Agent, which translates complex diagnostic output into a clear, actionable email, streamlining patient follow-up logistics. Most critically, the **Human-in-the-Loop** policy is enforced through the non-autonomous generation of the referral communication, making human validation of the AI's logic a mandated step before patient action.

The primary technical challenges involved ensuring robust **data compatibility** between the 6 parallel agents and the Multidisciplinary Team Aggregator. This was particularly evident in the necessity to enforce strict **JSON parsing** on the output of the MDT. Any deviation from the required JSON schema would have resulted in the failure to reliably extract the critical *specialist_recommendation* field, thereby halting the downstream Intelligent Referral step and undermining the action-oriented goal of the system. Overcoming these parsing inconsistencies was essential to guarantee the **Data Reliability** necessary for clinical application.

Bibliographic References

[1] C. Benker, E. Schlegel, A. Schmidt, and P. Fischer, “What Is Human-Centered AI? An Analysis of Definitions and the Path Forward,” ACM Trans. Comput.-Hum. Interact., vol. 29, no. 6, pp. 1–39, Dec. 2022.

[2] D. V. Voutsas and M. L. Louta, “The Role of Multidisciplinary Team in Decision Making of the Oncology Cases,” J. Med. Syst., vol. 45, no. 11, pp. 1–13, 2021.

[3] The AI Forum, “Implementing Agentic RAG using LangChain,” Medium, 2024.

[4] Ahmad VH, “AI Agents for Medical Diagnostics (GitHub Repository),” 2023.

[5] N. Bar, “AI Travel Agent Multi-Agent Travel Assistant (GitHub Repository),” 2023.